On the Complexity of Average Path length for Biological Networks and Patterns

Waqar Asif*[1] , Hassaan Khaliq Qureshi[1], Adnan Iqbal[2]and Muttukrishnan Rajarajan[3]

[1]School of Electrical Engineering & Computer Science (SEECS), National University of Sciences & Technology (NUST), Islamabad, Pakistan.
[2]Iqra University, Islamabad, Pakistan. [3] School of Engineering and Mathematical Sciences, City University, London, UK.

Email: Waqar Asif*- waqar.asif@gmail.com; Hassaan Khaliq Qureshi - hassaan.khaliq@seecs.edu.pk; Adnan Iqbal - adnan.iqbal@iqraisb.edu.pk; Muttukrishnan Rajarajan - r.muttukrishnan@city.ac.uk;

*Corresponding author

Abstract

Path length calculation is a frequent requirement in studies related to graph theoretic problems such as genetics. Standard method to calculate average path length of a graph requires traversing all nodes in the graph repeatedly, which is computationally expensive for graphs containing large number of nodes. We propose a novel method to calculate average path length for graphs commonly required in the studies of genetics. The proposed method is computationally less expensive and less time consuming compared to standard method. In this paper, a mathematical formulation is provided that calculates Average Path Length for graphs commonly present in biological networks, at the cost of uniform time complexity, for different size of networks.

Introduction

Over the past few years interaction networks have gained central importance in molecular biology, with the increase in amount of data being available due to high throughput methods. This has created the need for tailored analysis methods that can extract meaningful biological information from such complex data [1]. This information can be abstracted using graph theory, which is a branch of discrete mathematics. The principal of this theory is to categorize the problem under consideration into nodes (vertices, elements) and edges (connections). In biological networks, these nodes can be the system elements as proteins and genes,

and the interaction of these elements can be represented as edges. Edges in a graph can be directed when e_{ij} need not be equal to e_{ji}, identifying the source and target nodes of an interaction and undirected, when the direction of edge is immaterial or in other words $e_{ij} = e_{ji}$, representing a chemical bond between two atoms [2].

Abstracting biological networks as graphs provides us with an opportunity to take benefit from the advances in graph theory by using already developed tools and measures. A graph has several properties. One of the basic properties is the node degree, which represents the number of incident edges on a node. A molecule interacting with many distinct molecules has a high node degree for example Guanosine triphosphate (GTP) interacts with many proteins. Due to this reason, in signal transduction, a path could represent a sequence of interactions from signal sensing area to final target area. An averaged value of these path lengths would give an idea of the average number of interactions that take place from any sensing area to a target area in a network. This average is commonly known as the Average Path Length (APL).

APL is defined as the mean of the shortest path lengths between all pair of vertices and it represents the closeness and consequently, how quickly information transfer takes place in a network [3]. Most real world networks unexpectedly have short characteristic path lengths, as popularized by six degree of freedom play. This property is known as the small world property and is studied in detail in [4]. Most real networks are differentiated in being small world or ultra-small world network due to the behavior of their APL as logarithmic or double logarithmic scaling with network size n nodes [5]. For the complex biological systems, evolution should be under tight consideration as different proteins harbor different path lengths [6]. Therefore, APL is a natural network statistic to interpret and compute.

Since APL is an important metric, several formulas have been proposed to estimate the APL. The most commonly used method is to traverse a complete graph and then average out all the path lengths to calculate its APL. It is denoted as:

$$l = \frac{1}{n(n-1)}[\sum_{i,j} d(v_i, v_j)], \tag{1}$$

Where n is the number of vertices/nodes in the network and d is the distance between nodes i and j for all pair of nodes in the network. For large sized networks this is quite non-trivial, hence, for simplicity Fronczak et al. in [7] used the hidden variable network model generalized in [8] to derive a formula for the average distance between each pair of nodes which was characterized by the given values of hidden variables h_i and h_j. They attained a good agreement but only for dense networks. Likewise Zhongzhi et al. in [5] have

derived a formula for APL characterization for the Apollonian network. Their analytical method is based on a recursive construction and a similar structure of Apollonian network. They have provided rigorous results showing that APL grows logarithmically with the number of nodes. This result is in contradiction with Jose et al. who state in [9] that Apollonian networks scale sub-logarithmically with the network size.

Similarly, numerous models have been proposed for networks with small average distance [10] [11] such as the static Watts-Strogatz model, in which a small percentage of edges are changed in a low dimensional lattice [12] or dynamic models, in which distance between nodes becomes smaller as more nodes are added to the network. Philippe et al. in [13] provide a closed form formula for an upper bound on APL for a recently proposed recursively growing network in [3].

All these proposed formulations focus on determining APL for recursively growing networks. To the best of our knowledge, none of the algorithms have focused on determining the APL of any arbitrary network at any given instance without knowing the prior growth trend. This limits researchers from knowing the actual APL for networks such as, biological networks where slight mistake in the approximation can mislead to incorrect protein folding structure and hence, a different biological structure [14].

Common Protein Structures

Reductionism has helped researchers over a century in deeply analyzing the functional characteristics of many cellular components. Despite this fact, this phenomenon faces immense challenges when used in biological networks, as it is rarely possible to attribute discrete biological functions into individual molecules. Hence, the challenge lies in understanding the dynamics and structure of a complex inter-cellular interaction that can be attributed as a function of a living cell.

One of the most extensively analyzed biological networks is a Protein-Protein Interaction (PPI) network. In PPI network each protein is treated as a node and an interaction as an edge. A PPI network exhibits the so called 'power law' distribution [15], where most proteins have a few neighbors and only a few have a large number of neighbors. Proteins with a large number of neighbors are known as 'hubs' as shown in Fig 1. It is proposed that hubs in a PPI network evolve slower than non-hub nodes and have higher importance. These proposals have inspired numerous in-depth studies to evaluate the position of proteins in a PPI network [16]. These studies often examine the correlation between a network characteristic and a biological variable, whereas it is highly appreciated that it should be confirmed that this measure is not due to some confounding effect of other variables. Thus, network approach when combined with a cautious statistical treatment of biological variables can yield an important insight into the properties of biological

networks.

The three dimensional structure of proteins is the key for understanding their evolution and functions. It is often stated that folding of proteins to its native state is the second genetic code [14]. Therefore, careful analyses of stable three dimensional folded structures of proteins provide insight into their functions and folding stability. Due to this reasons, results of analysis have become the foundation of protein structure prediction from amino acid sequence. Many structures can differ in terms of conformational features (geometry) but still can have the same topology (the gross shape), hence, precise calculations are required.

Protein structures have geometry expressed in the form of protein backbone & side chains and this can be described in terms of graph theory as a tree structure as shown in Fig 2, which represents a PPI in Saccharomyces cerevisiae. It is based on early yeast two hybrid measurements and illustrates a few highly connected nodes, which hold the network together. The color of a node indicates the phenotypic effect of removing a corresponding protein (green=non-lethal, red=lethal, yellow=unknown and orange=slow growth) [17]. For such complex network slight mistake in calculating the APL can mislead to an incorrect protein folding structure.

In this paper, we present a new method to calculate APL for graphs, which are frequently encountered in biological networks. The proposed method does not require traversing the complete graph and is computationally less expensive.

Average Path Length Calculation for Common Protein Structures

Average Path Length (APL) of a graph l is defined as the average number of edges along the shortest path for all possible pairs of network nodes and is represented using eq 1. For any general graph, calculating l requires traversing a complete graph several times. However, if the graph is a regular one - as many of the biological networks are – then simpler and less time consuming methods can be found.

We describe one such method here for biological networks of PPI, shown in Fig 2. We use a divide and conquer approach by first modeling individual sub-parts of a network and then joining them together to get a unified formula.

To start with, we assume that the network is a line graph with n vertices as shown in Fig 3a. This implies that there are $n-1$ edges in the network and the maximum degree is 2. This information is sufficient to calculate average path length l of such a network denoted as:

$$l = \frac{2n-1}{6} + \frac{1}{2}, \tag{2}$$

The equation above greatly simplifies the process of calculating APL for simple line graphs and it requires only one parameter – number of nodes on the network n.

We now consider a star shape network with one central node and $n-1$ nodes connected to it directly, as shown in Fig 3c. In such networks, central node has the highest degree $n-1$. For these networks l can be written as:

$$l = \frac{2(n-1)}{n}, \tag{3}$$

Just like eq 2, this is also dependent on only one parameter i.e., number of nodes in the network. It is now considered that both of these simple graphs are combined in such a way that they partially form a line graph and a star graph, as shown in Fig 3b. In parallel, it is considered that two parameters from these type of graphs are known, namely 1) the number of nodes in the graph n and 2) the highest degree of the graph d. For such graphs, APL can be written as:

$$l = \frac{1}{n(n-1)}(n-d+1)(n-d+2) \tag{4}$$
$$[\frac{1}{2}(\frac{(2n-2d+3)}{3}+1)+(d-2)]+2[(d-1)(d-2)],$$

The above equation provides a generalized expression for tree graphs shown in Fig 3b. Some important steps in the derivation of above equations are mentioned in the Appendix. Through eq 4, having known only two parameters, number of nodes (n) and highest degree (d), we can calculate the APL. APL can be calculated now without traversing whole graph repeatedly, thus saving computation time and memory required for expensive graph storage and search operations. In next section, we discuss in some detail the derivation of eq 4.

Complexity Analysis

A common time complexity evaluation method is to use asymptotic worst case analysis and to express the computational cost as a function dependent on the size of the input. However, for an incremental algorithm that is in result of a dynamic problem, such an analysis is not very informative [18]. When the cost of computation is expressed as the size of the current input, several proposed algorithms running in time asymptotically are no better than the time required for performing the computation from scratch. An incremental algorithm is said to be bounded, if a function exists that can bound the change in size of the input and output by the time taken in performing an update. An incremental algorithm is said to be

unbounded with the model of comparison if it does not have a bounded incremental algorithm within that model of computation.

The complexity of a bounded algorithm depends on the degree to which the set of vertices with unchanged value connect to the set of vertices with changed values [18]. This results in a constant change in complexity with the increase in the size of a network. This means, for large sized growing networks computation of APL at any given instance gets more complex with the increase in size for conventional APL calculating recursive algorithms. For validation of our results we used the Network Analyzer tool in Cyctoscape [19] to generated a tree structure and evaluated the time taken for calculation of APL. Fig 5 shows that as the numbers of nodes in a graph increase, the time taken by the Network Analyzer tool increases exponentially. Hence, with the use of these techniques, APL calculation for a large sized, complex biological network can be time consuming.

On the other hand, method proposed in this paper determines APL with constant time complexity. Our proposed mathematical formulation determines the APL based on the number of nodes and the highest degree hence it has uniform complexity for all size of graph.

Conclusion

In biological networks, different proteins harbor different path lengths. Many PPI structures can have the same topology and yet differ in characteristics in terms of conformational features (path length). Hence, careful APL calculations are required to monitor unique protein folding patterns. Traditional APL calculating techniques monitored the growth trend of a network for determining a close approximation. These techniques suffered from increased complexity with the increase in network size. In this paper, a mathematical formulation is provided that calculates APL for graphs commonly present in biological networks. In addition, the time complexity for calculating the APL does not increase with the increase in the network size.

Appendix

APL is defined as the mean of the shortest path lengths of all the nodes in a graph and is denoted using eq 1. The averaging factor $1/n(n-1)$ remains same for all set of graphs, hence, we focus on calculating path lengths for different graphs and finally conclude by deriving eq 4 which satisfies all set of graphs.

If we have a line graph, as shown in Fig 3a, with n nodes than for non-edge nodes we have paths in two directions the upper side of graph and the lower side of graph. If we move from top to bottom or vice versa

than we have:

$$l = \frac{1}{n(n-1)}(\sum_{i=1}^{n-1} d(i+0) + \sum_{i=1}^{n-2} d(i+1) + \sum_{i=1}^{n-3} d(i+2) \\ .... + \sum_{i=1}^{n-1} d(i) + \sum_{i=1}^{n-n} d(i)), \tag{5}$$

While traversing the complete graph the average distance of a node replicates itself after we cross the center of the line graph resulting in:

$$l = \frac{1}{n(n-1)} 2(\sum_{i=1}^{n-1} i + \sum_{i=1}^{n-2} i + \sum_{i=1}^{n-1} i + ... + \sum_{i=1}^{2} i + \sum_{i=1}^{1} i), \tag{6}$$

The summation function ($\sum_{i=1}^{n-1} i$) is an arithmetic series and can be simplified as $\frac{n(n+1)}{2}$, hence

$$l = \frac{1}{n(n-1)} 2(\frac{(n-1)(n-1+1)}{2} + \frac{(n-2)(n-2+1)}{2} \\ .... + \frac{2(2+1)}{2} + \frac{2(1+1)}{2}), \tag{7}$$

Simplifying eq 7 we have:

$$l = \frac{1}{n(n-1)}(n(n-1) + (n-1)(n-2) + ... + 6 + 2), \tag{8}$$

Eq 8 can further be simplified to:

$$l = \frac{1}{n(n-1)} \sum_{i=1}^{n-1} (i)(i+1), \tag{9}$$

By expanding this we have our final form for the APL of a line graph and that is:

$$l = \frac{1}{n(n-1)}(\frac{n(n-1)(2n-1)}{6} + \frac{n(n-1)}{2}), \tag{10}$$

Similarly for finding the APL for a tree graph, we can divide a tree into three parts: a line graph that starts from the top of the tree to the lowest end of a tree, the second part calculates the distance of all the leaf nodes connected to the highest degree node and the third part calculates the distance of all the nodes connected to the highest degree node with each other.

Part I

The path length of the line graph shown in Fig 4 can be calculated using eq 9, but here we need to eliminate all the leaf nodes except one, hence we have:

$$l = \frac{1}{n(n-1)}(\sum_{i=1}^{n-(d-1)} i + \sum_{i=1}^{n-(d-1)} i^2), \tag{11}$$

By expanding eq 11 we have:

$$l = \frac{1}{n(n-1)} \frac{(n-d+1)(n-d+2)}{2}(1 + \frac{2n-2d+3}{3}), \tag{12}$$

Part II

In this part, we calculate the path length of all the leaf nodes of the tree to all the non-leaf nodes in a graph as shown in Fig 4. For this purpose, we sum up the path lengths of all the leaf nodes except the leaf nodes covered in *Part I*. This part varies from the first part in a way that in *Part I*, we also include the distance between all the nodes in the line whereas here we calculate, the path length of any single node (that is part of the line graph) to all the leaf nodes except the one considered in *Part I*. Starting from the farthest node in the line graph and moving towards the lower part (leaf nodes), we sum up the path lengths of all the non-leaf nodes with only the leaf nodes. The path length between all the non-leaf nodes is calculated in *Part I*, hence, we do not calculate it in this part. The path length for this part can be represented by:

$$l = \frac{1}{n(n-1)}[2(d-2) \sum_{i=1}^{n-(d-1)} i], \tag{13}$$

Part III

In this part, we calculate the inter leaf nodes path length for the graph shown in Fig 4. The distance between any two leaf nodes is two units and the total number of such combinations can be represented by:

$$l = \frac{1}{n(n-1)}(2(d-1)(d-2)), \tag{14}$$

APL for a tree graph

To find the APL for a tree graph we sum up eq 12, 13 and 14. Now we have:

$$l = \frac{1}{n(n-1)} \frac{(n-d+2)(n-d+1)}{2}[\frac{2n-2d+3}{3} + 1] + \tag{15}$$

$$2(d-2)\sum_{i=1}^{n-(d-1)} i + 2[(d-1)(d-2)],$$

By solving eq 15 we have our generalized equation for calculating the path length of any given graph:

$$l = \frac{1}{n(n-1)}(n-d+1)(n-d+2), \tag{16}$$
$$[\frac{1}{2}(\frac{(2n-2d+3)}{3}+1)+(d-2)]+2[(d-1)(d-2)],$$

By simply inserting the values of the highest degree and the total number of nodes, one can find the APL for any arbitrary graph at any instance. In the next section, we discuss in some detail how much time and memory can be saved.

References

1. Mason O, Verwoerd M: **Graph theory and networks in biology**. *Systems Biology, IET* 2007, **1**(2):89–119.
2. Gursoy A, Keskin O, Nussinov R: **Topological properties of protein interaction networks from a structural perspective**. *Biochemical Society Transactions* 2009, **36**(6):1398.
3. Zhang Z, Rong L, Guo C: **A deterministic small-world network created by edge iterations**. *Physica A: Statistical Mechanics and its Applications* 2006, **363**(2):567–572.
4. Strogatz S: **Exploring complex networks**. *Nature* 2001, **410**(6825):268–276.
5. Zhang Z, Chen L, Zhou S, Fang L, Guan J, Zou T: **Analytical solution of average path length for Apollonian networks**. *Physical Review E* 2008, **77**:017102.
6. Xu K, Bezakova I, Bunimovich L, Yi S: **Path lengths in protein–protein interaction networks and biological complexity**. *Proteomics* 2011.
7. Fronczak A, Fronczak P, Hołyst J: **Average path length in random networks**. *Physical Review E* 2004, **70**(5):056110.
8. Serrano M, Boguná M: **Topology of the world trade web**. *Physical Review E* 2003, **68**:015101.
9. Andrade Jr J, Herrmann H, Andrade R, da Silva L: **Apollonian networks: Simultaneously scale-free, small world, Euclidean, space filling, and with matching graphs**. *Physical review letters* 2005, **94**:18702.
10. Schnettler S: **A structured overview of 50 years of small-world research**. *Social Networks* 2009, **31**(3):165–178.
11. Giabbanelli P: **Impact of complex network properties on routing in backbone networks**. In *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, IEEE 2010:389–393.
12. Watts D: **Networks, dynamics, and the small-world phenomenon 1**. *American Journal of Sociology* 1999, **105**(2):493–527.
13. Giabbanelli P, Mazauric D, Bermond J: **On the average path length of deterministic and stochastics recursive networks**. *Complex Networks* 2011, :1–12.
14. Vishveshwara S, Brinda K, Kannan N: **Protein structure: insights from graph theory**. *Journal of Theoretical and Computational Chemistry* 2002, **1**:187–212.
15. Jeong H, Tombor B, Albert R, Oltvai Z, Barabási A: **The large-scale organization of metabolic networks**. *Nature* 2000, **407**(6804):651–654.
16. Li M, Wang J, Wang H, Pan Y: **Essential proteins discovery from weighted protein interaction networks**. *Bioinformatics Research and Applications* 2010, :89–100.
17. Barabási A, Oltvai Z: **Network biology: understanding the cell's functional organization**. *Nature Reviews Genetics* 2004, **5**(2):101–113.
18. Ramalingam G, Reps T: **On the computational complexity of dynamic graph problems**. *Theoretical Computer Science* 1996, **158**(1-2):233–277.
19. **Cytoscape** [http://apps.cytoscape.org/apps/networkanalyzer].

Figures

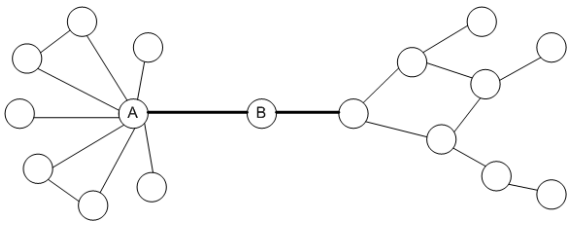

Figure 1: Protein interaction network: a simple graph of a protein interaction network. Node A is a hub, node B is a non-hub [17].

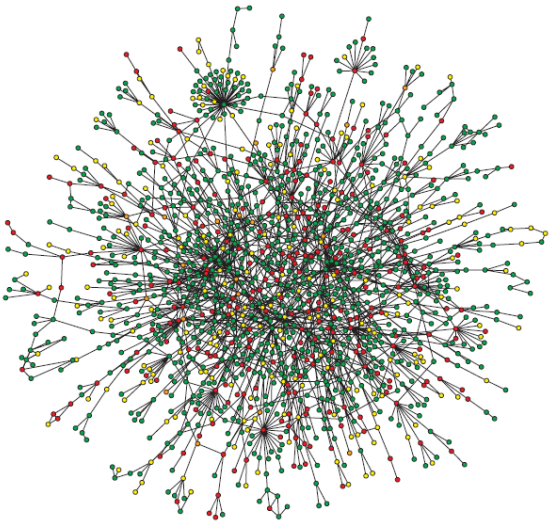

Figure 2: Protein interaction network: A map of protein-protein interactions based on early yeast two hybrid measurements. It illustrates that a few highly connected nodes hold the network together [17].

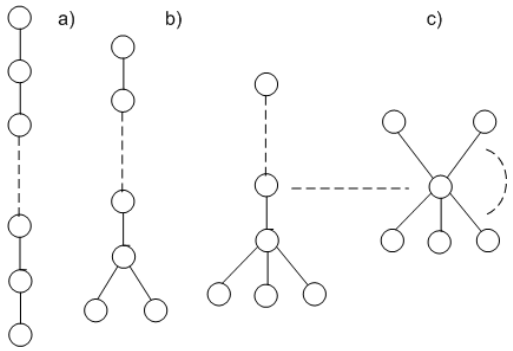

Figure 3: A reduced constructed topology from a line graph to a star graph.

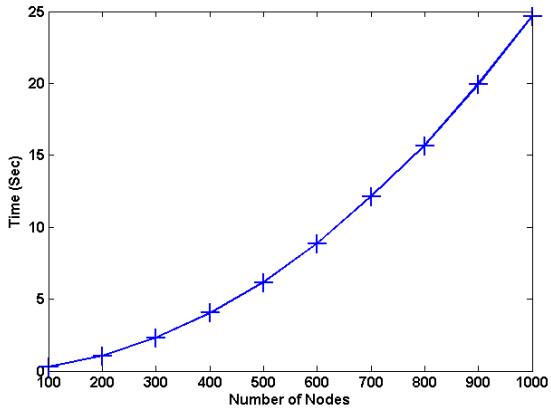

Figure 4: Time required to calculate APL for conventional algorithms.

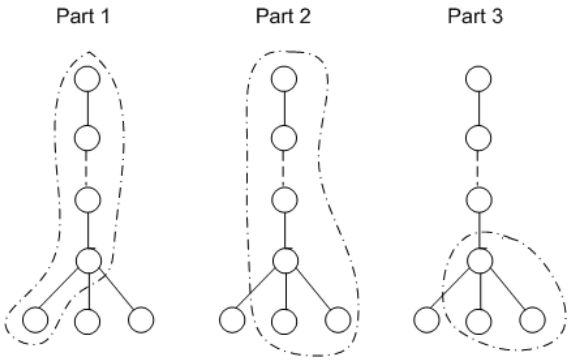

Figure 5: Tree graph further divided into three parts.